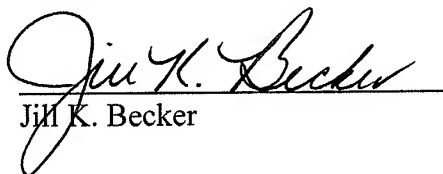


**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Applicants: Curran et al. Confirmation No.: 2326  
Serial No.: 10/730,508 Group Art Unit: 2151  
Filed: December 8, 2003 Examiner: Glenford J. Madamba  
Title: DATA MOVEMENT MANAGEMENT SYSTEM AND METHOD FOR A  
STORAGE AREA NETWORK FILE SYSTEM EMPLOYING THE DATA  
MANAGEMENT APPLICATION PROGRAMMING INTERFACE

**CERTIFICATE OF ELECTRONIC TRANSMISSION**

I hereby certify that this correspondence is being transmitted electronically to: Mail Stop Appeal Brief – Patents, United States Patent and Trademark Office, Alexandria, VA 22313-1450 on July 18, 2008.

  
Jill K. Becker

Date of Signature: July 18, 2008

To: Mail Stop Appeal Brief-Patents  
P. O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF SUBMITTED UNDER 37 CFR § 41.67**

Dear Sir:

This is an Appeal from a Final rejection dated February 20, 2008. A response under 37 CFR § 1.116 was submitted on April 18, 2008. A Notice of Appeal was submitted by the Applicants on May 20, 2008. This brief is therefore due prior to July 20, 2008 and is thus seen to be timely submitted. The presently submitted brief is accompanied by an authorization to charge Appellants' Assignee's deposit account in the amount required by 37 CFR § 1.17(c).

## **I. REAL PARTY IN INTEREST**

The real party in interest is International Business Machines Corporation by virtue of an assignment agreement executed on December 3, 2003 and December 8, 2003 by the inventors herein and recorded on March 12, 2004 in the U. S. Patent and Trademark Office on Reel 014421 and Frame 0840.

## **II. RELATED APPEAL AND INTERFERENCES**

To the knowledge of the undersigned, Appellants' attorney, there are no other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

## **III. STATUS OF CLAIMS**

Claims present: 1-20

Claims allowed: none

Claims rejected: 1-20

Claims objected to: none

Claims cancelled: none

Claims appealed: 1-20

## **IV. STATUS OF AMENDMENTS**

Rule 116 amendment to claims: none presented, all amendments previously entered.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

### **Summary of Independent Claim 1**

A method of managing data movement is provided in which a processing environment [100 in Figures 1 and 2] is established in a cluster [100] of nodes [110a -110n]. See Applicants' specification paragraph [0022]. The nodes [110] have common access [130 and 230 in Figures 1 and 2] to data residing in one or more data storage units [120]. See Applicants' specification paragraph [0024]. A data management application (DM) is initiated in the environment [Specification paragraph [0033]]. **It is to be particularly noted**

**that the parenthetical present in applicants claims is not intended to be superfluous.** It is intended to invoke the DM API standard. One of the nodes of the cluster is assigned [e.g., 110a as described in paragraph [0034]] as a coordinating node for managing data movement [520 in Figure 5 and paragraph [0034] in Applicants' specification]. Thereafter, a worker thread is posted [530 in Figure 5] to one or more of the nodes in the cluster to perform one or more data movement tasks [556 in Figure 5] in response to the event. The posting of worker threads to carry out the data movement request is not an activity described in any of the art cited.

#### **Summary of Independent Claim 19**

This claim is a "program product" version of claim 1. A computer readable medium is provided with instructions for carrying out the method which is described in Applicants' claim 1. In particular, these instructions provide a method for managing data movement in a processing environment [100 in Figures 1 and 2] established in a cluster [100] of nodes [110a-110n]. See Applicants' specification paragraph [0022]. The nodes [110] have common access [130 and 230 in Figures 1 and 2] to data residing in one or more data storage units [120]. See Applicants' specification paragraph [0024]. A data management application (DM) is initiated in the environment [Specification paragraph [0033]]. One of the nodes of the cluster is assigned [e.g., 110a as described in paragraph [0034]] as a coordinating node for managing data movement [520 in Figure 5 and paragraph [0034] in Applicants' specification]. Thereafter, a worker thread is posted [530 in Figure 5] to one or more of the nodes in the cluster to perform one or more data movement tasks [556 in Figure 5] in response to the event.

#### **Summary of Independent Claim 20**

Claim 20 is directed to a data processing system [50 in Applicants' Figures 1 and 2 and in paragraph [0022]] which includes program instructions for carrying out the method which is described in Applicants' claim 1. The claimed system manages data movement. A computing environment [50] has a cluster of nodes [110] having common access to data residing in one or more data storage units [120]. A data management application (DM) [520 in Figure 5 and paragraph [0034] in Applicants' specification] operable to manage data

movement assigns [e.g., 110a as described in paragraph [0034]] any node in the cluster the role of coordinating node to manage data movement events. This coordinating node dispatches worker threads [530 in Figure 5] to one or more nodes to perform data movement tasks in response to data movement request events [556 in Figure 5].

## **VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

**Issue #1:** Whether Applicants' affidavit submitted under 37 CFR § 1.131 is sufficient.

**Issue #2:** Whether Claims 1- 20 are anticipated under 35 USC §102(e) based upon the published patent application of Moore et al. (U.S. Patent Publication US 2004/0249904 A1, having a publication date of December 9, 2004 and bearing a filing date of April 16, 2003, hereinafter sometimes referred to simply as Moore).

**Issue #3:** Whether Claims 5-8 are rendered obvious under 35 USC § 103 based upon the aforementioned Moore application in further review of the published patent application of Dugan et al. (U.S. Patent Publication US 2006/0165223 A1, having a publication date of July 27, 2006 and bearing a filing date of February 15, 2006, hereinafter sometimes referred to simply as Dugan).

## **VII. ARGUMENTS**

### **Issue #1 – Affidavit Sufficiency**

The Examiner has also asserted that the previously submitted affidavit under 37 CFR § 1.131 is inadequate. Applicants vehemently disagree with this assertion.

When an applicant's invention is directed to a device such as a golf tee or a Frisbee, a successful actual reduction to practice occurs when the device has been produced, used and found to perform its desired function.

No less can or should be said when the applicant's invention is directed to a method for operating a data processing system. In such a case, the production of the device corresponds to the coding and compilation of the necessary software. The use of such an invention occurs when the software is executed on a data processing system. A successful

execution in such cases results in the transfer of data in the manner called for by the program. No better indication of a successful test can be found than an indication that the program has been approved for sale as a licensed product.

This is how software is developed. If this were an invention related to a golf tee for example, an inventor could take a picture of the resultant device together with evidence that the picture was taken on a particular date. Such evidence as this has appeared in inventors' notebooks has been well accepted for decades as sufficient proof under Rule 131 affidavits. In this particular instance, the result of a successful test of the invention was the movement of data in the manner described in the claims at issue, namely through the posting of one or more working threads for carrying out the transfer. Accordingly, it is seen that in the software arts the possibility of producing a photograph of the method is not possible. In the software arts, code is written and applied against a large plurality of scenarios and conditions to test its workability. However, there is nothing within the regulations, the statutes or the judicial interpretations of these codes which would in any way indicate that software related inventions would be or should be treated any differently under Rule 131.

It is and has been a long-standing principle of patent law that a successful actual reduction to practice is seen to be present when the inventors themselves perceive a test of the invention to be successful.

It is suspected that as a negative reaction to the affidavit submitted herein under Rule 131 is based upon a lack of knowledge, understanding or consideration of the process for developing software. Accordingly, it is Appellant's belief that proper weight is not being given to the use of the terms "unit test," "functional test," or "code library." In particular, it is noted that the Wikipedia website (as viewed on July 13, 2008) defines "unit test" as follows: "In computer programming, unit testing is a test that validates that individual units of source code are working properly." Accordingly, when an affiant says that unit tests have been passed, it is an indication that the code is indeed functioning as intended. Likewise, the same website, at the same time, indicates that code libraries "...contain code and data that provide services to independent programs." Thus to say that code is placed in a library means that it is indeed capable of providing the indicated services. To put this slightly

differently and perhaps in a more understandable fashion, standard public libraries do not stock their shelves with books that authors have not finished writing. There is a process of vetting carried out by publishers before books are placed on library shelves. Thus to say that something is in a library, especially a code library, implies that it is suitable for its intended purpose. It is furthermore asserted that the expression “functional testing” is not being afforded its full and intended meaning. Again, it is noted that the Wikipedia website asserts that functional testing is testing is performed on software prior to its delivery in the following language: “In engineering and its various subdisciplines, acceptance testing is black-box **testing performed on a system** (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) **prior to its delivery**. In some engineering subdisciplines, it is known as **functional testing**, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, usability testing, or factory acceptance testing.”

It is therefore Applicants’ position that full and proper consideration has not been given to the language, content and meaning present in the submitted affidavit. Furthermore, it is Applicants’ position that they have provided all that is reasonably possible to submit as evidence of a successful reduction to practice in the software arts, at least with respect to this invention.

With respect to the latter point, it is to be noted that, even if it were practical to submit a picture of the bit patterns on the disks that occurred as a result of the claimed process, there would still be no indication that that information was written to those disks in a parallel fashion. Nonetheless such a limitation should not deprive inventors in the software arts of the opportunity of submitting affidavits under Rule 131.

In any enterprise which commercially produces a software product there is a process for vetting that software. That process can be likened to business records production. Accordingly, the present Applicants assert that the submitted affidavit readily references a well understood a software vetting process wherein the results of that process is a successful test of that software. In software testing, the proof of the pudding, so to speak, is the passage of that software to a stage of commercialization which by itself is indicative of a successful

completion of software test. In short, the present Applicants have set forth a process by which all software is tested, not just by the present assignee, but by every significant commercial software producer. Put another way, the present Applicants have demonstrated that there is a process for maintaining a business record pertaining to software development, testing and completion. Furthermore, Applicants' affidavit clearly sets forth the fact that the present software was tested and completed and that records of this completion were created in accordance with a well-established business records rule.

Accordingly, it is seen that, in the context of software development, the presently submitted affidavit is ample evidence of a successful completion of the invention prior to the date indicated. It is therefore requested that the affidavit be considered to be fully indicative of a successful proof of concept and a successful reduction to practice. Considered as such, it is seen that the reduction of practice of the present invention prior to the priority date of the patent application of Moore is more than sufficient to remove it as a reference in the present prosecution. For this reason also it is seen that the rejection of Applicants' claims based upon the cited published patent application of Moore be withdrawn.

For all the reasons asserted above, it is Applicants' position that the rejection of claims 1-4 and 9-20 under 35 USC § 102(e) based upon the published patent application of Moore et al., cannot be sustained. It is therefore requested that it be withdrawn for reasons based upon the sufficiency of the affidavit.

#### **Issue #2 – Rejection under 35 USC § 102(e)**

With respect to the rejection under 35 USC § 102, the Board is reminded of the fact that rejections under this statute constitute a narrow ground of rejection. The so-called anticipation rejection **requires each and every recited claim element to be found** within the four corners of a single document. Any of the minor exceptions to this rule are not germane to the present discussion.

Attention is now focused on a discussion of the differences found between the patent application of Moore and the subject claim language. For everyone's convenience, the allegedly identical elements are set forth below in an abbreviated table (Table I). Additionally, also for the convenience of the Board, a second table (Table II) is also

provided below in which focus is directed to the differences between Applicants' recited claim elements and the cited portions from the patent application to Moore et al. upon which the Examiner relies. Likewise, for everyone's additional convenience, it is to be particularly noted that the presently claimed invention and the method of Moore et al. differ in several respects, but most prominently in the last recited claim element, namely the step of "posting a worker thread to **one or more of the nodes** to perform data movement in response to the event."

<b><u>Applicant's claim 1</u></b>	<b><u>Cited portion from Moore</u></b>
establishing a processing environment in a cluster of nodes.....	A cluster of computer system nodes connected by a storage area network include two classes of nodes. The first class of nodes can act as clients or servers, while the other nodes can only be clients.
initiating a data management application...	Furthermore, conventional hierarchal storage management uses an industry standard interface called data migration application programming interface (DMAPI). However, if there are five machines, each accessing the same file, there will be five separate events and there is nothing tying those DMAPI events together.
<b>assigning a node of said cluster as a coordinating node...</b>	<b>[Figure 5-7]</b>
receiving an event by the coordinating node...	The possible DMAPI events are read, write and truncate.
<b>posting a worker thread...</b>	<b>Reference to paragraphs [0103-0105] and [0117-0118].</b>

**Table I**



<b><u>Applicant's claim 1</u></b>	<b><u>Reasons for Difference</u></b>
establishing a processing environment in a cluster of nodes.....	Moore et al. divided as their processing environment into a system with only client nodes and server nodes. There is no such required a dichotomy presents in applicants claimed method. It is also to be particularly noted in this regard that applicants recitation of a session that node is the recitation of something that is different from either a client node, a server node or a meta data node.
initiating a data management application...	Here the only relevance to be found within the site its patent application is the fact that Moore et al. also refer to the data migration application programming interface (DMAPI). Moreover it is noted that in the cited patent application there is only a reference to distinct events and does not in any way refer to a single event being processed in a parallel fashion. In contrast, it is seen that applicants claimed invention "looks below" the event processing request level. Applicants' invention provides a view interior to step 138 in the application of Moore et al.
<b>assigning a node of said cluster as a coordinating node...</b>	<b>The cited patent application does not in any way appreciate the concept of a coordinating node as that term is used in Applicants' specification and claims.</b>
receiving an event by the coordinating node...	The possible DMAPI events are read, write and truncate...
<b>posting a worker thread...</b>	<b>This aspect of Applicants' claims is nowhere taught, disclosed or even remotely suggested by the cited application. The posting of worker threads to one or more nodes clearly indicates that the event processing for data migration has the capability of being carried out in a parallel fashion, a clear vantage over the art which lacks this capability.</b>

**Table II**

In consideration of this position, the Board's attention is directed to the following language from paragraph [0010] from Applicants' specification: "In these filings [referring to earlier filed related patent applications assigned to the same assignee as herein], the use of the DMAPI standard is provided **without modification**. However, all data migration and recall is conducted **through a single node called a session node**." Attention is further directed to the following language from paragraph [0011] from Applicants' specification: "Consequently, **it would be desirable to utilize multiple nodes for data movement** under coordination of a DMAPI application on a single session node to enhance performance

without altering the operating system, the components of the computing environment or the DMAPI standard.” [All emphasis added herein.]

It is very important that the Board understand the context of the present invention with respect to the patent to Moore. **In point of fact, the present invention provides a better way of handling step 138 of Figure 7 of the Moore et al. application and as described in associated paragraph [0076] of their application.** The present invention does more than just process a DMAPI event. It processes it in a specific fashion, namely through the posting of worker threads on multiple nodes thus permitting DMAPI events to be processed more quickly, and ultimately more reliably. It is also to be particularly noted that it is important for the board and to the examiner to realize that, in the present context, **the expression “data management” is a term of art.** It applies to the management of hierarchically stored data. In support of this position, apart from the information already provided in Applicants’ specification, is noted that the Wikipedia website provides a consistent definition for this term particularly as it pertains to the DM API standard: “Data Management API (DMAPI) is the interface defined in the X/Open document "Systems Management: Data Storage Management (XDSM) API" dated February 1997. XFS, IBM JFS, VxFS, AdvFS and GPFS file systems support DMAPI for Hierarchical Storage Management.”

Attention is particularly directed to Figure 7 of the cited patent application (which, incidentally, the Examiner relies upon to support his position). In this drawing it is seen that **there are only two nodes shown**, a client node and a server node. **However, it is absolutely clear from this drawing that the only processing of the indicated DMAPI event occurs in their server node.** Moore et al. limit DMAPI event processing for a given file system to their metadata server node which gives them the same problem that led the present inventors to the method described in the present application. Accordingly, it is seen that Moore et al. utterly fail to appreciate the idea that data movement **in the DMAPI context** may be carried out by more than one node **in response to a DMAPI request event.** Applicant’s specification points out the fact that this is not an easy thing to accomplish. In particular, the Board’s attention is directed to paragraph [0009] below from Applicant’s specification:

“Incorporation of DMAPI into SAN environments is **challenging** and often involves undue restrictions. Prior approaches to incorporating DMAPI in SAN environments have limited data accessibility by requiring use of a mirror server, thus affecting performance and adding cost and complexity to data processing. In other approaches, the prior art requires changes to the running operating system or even to the DMAPI standard itself, both of which are undesirable. Neither such approach is desirable.”

With respect to indicated paragraphs [0103-0105] and [0117-0118], there is no data movement within the context of a DMAPI event anywhere within the cited lines. Even more particularly, with respect to paragraph [0117] it is seen that it and subsequent paragraph [0118] are solely directed to recovery and relocation: “Preferably interruptible token acquisition is used to enable **recovery and relocation** in several ways.” They have nothing whatsoever to do with responding to DMAPI event requests or to the posting of threads. While the cited material mentions the use of threads, it is only their use of this single word that is at all relevant. However, there’s actually nothing in the cited material that refers to the use of threads as being used to respond to DMAPI event requests. Furthermore, there is nothing in the cited material which indicates or even remotely suggests the posting of multiple threads to respond to such events. As pointed out in Applicants’ specification, the art regards such efforts as being “challenging.”

As indicated above in the summary of Applicants’ claim 1, the expression “(DM)” is intended to have a specific meaning. It cannot be regarded as merely superfluous language. Furthermore, it is present in the other ones of Applicants’ independent claims as well. Is intended to invoke the context of the DMA API standard. **Based upon the Examiner’s prior responses, it is clear that no patentable weight has been given to this phrase.**

### **Issue #3 – Rejection under 35 USC § 103**

Attention is now directed to the other art based rejection imposed by the Examiner. In particular, claims 5-8 also stand as being rejected under 35 USC § 103 based upon the aforementioned Moore et al. application in further review of the published patent application of Dugan et al.

It is initially noted that if the Rule 131 affidavit submitted previously is indeed found to be sufficient, then this rejection also must fail since the acceptance of this affidavit removes the-based patent upon which the rejection is based. It is applicant's assertion that this should indeed be the case. Nonetheless, it is still incumbent upon the Applicants to consider the alternative.

It would appear that it would be very useful to begin this discussion with a proper understanding of the term "session." A "session" is a specific DMAPI concept defined in the standard. Prior patents, assigned to the same assignee as the present invention in this area describe this in detail. See explicitly the following quote from U. S. Patent 6,990,478: "The node on which the session is created is designated as the session node, and all specified events generated by file system operations are reported to the session node, regardless of the node at which the events are generated." The Dugan's concept, and correspondingly the Examiner's concept, of a "session" is taken from the network world rather than the data management world. They simply mean entirely different things.

Furthermore, the Dugan application is a networking application which the Examiner appears to be citing solely because all of their use of the word "session." However, as pointed out above, this term has dual meanings in the computer arts. These disparate meanings are described above but it is noted that, in the present context, as used by a Dugan, the word "session" has nothing at all to do with data management which is the subject of the present invention.

With specific reference to claim 5, it is noted that the word "session identifier" is to be found therein. It is noted that Moore et al. also appeared to teach the use of a session identifier. However, they do so only because the standard requires one. Apart from that it has no use in the cited patent application of Moore et al. The added patentable value set forth in claim 5 is that the session identifier has meaning on multiple threads on multiple nodes. **Nothing in Moore et al. goes in that direction and Dugan has a different concept of session.**

Furthermore, with respect to claim 5, it is noted that it depends directly from claim 1. Since it depends from claim 1 it therefore contains a recitation directed to "posting a worker

thread to one or more of the nodes to perform data movement in response to the event.”

**Furthermore, although not specifically emphasized above, it is noted that the posting of the thread is a step carried out in response to a very specifically identified event. It is an event from the coordinating node requesting movement of data.** As such, Applicants’ claim 5 contains concepts and ideas that are missing from both of the cited documents. In this regard, it is noted that it is a well-settled principle of patent law that even in a combination rejection under 35 USC § 103 is still incumbent upon the Examiner to either find the recited claim element within the totality of the art cited or to provide clear evidence that such a claim element is at least suggested. The Examiner has failed to do this in the present situation simply because it is not possible. The recited concepts are simply not present. The only similarity between the art cited in Applicants’ claimed invention lies in a collection of search buzzwords.

For all the reasons stated above, it is seen that the rejections of Applicants’ claims 1-20, as set forth above cannot be sustained. It is therefore very respectfully requested that the rejection of Applicants’ claims be reversed.

Very respectfully submitted,



---

Lawrence D. Cutter  
Attorney for Appellants  
Registration No. 28,501

Dated: July 18, 2008

HESLIN ROTHENBERG FARLEY & MESITI P.C.  
5 Columbia Circle  
Albany, New York 12203-5160  
Telephone: (518) 452-5600  
Direct telephone: (845) 255-3276  
Facsimile: (518) 452-5579

## VIII. CLAIMS APPENDIX

1. A method of managing data movement, comprising:
  - establishing a processing environment in a cluster of nodes having common access to data residing in one or more data storage units;
  - initiating a data management application (DM) in said environment;
  - assigning a node of said cluster as a coordinating node for managing data movement;
  - receiving an event by the coordinating node requesting movement of data;
  - posting a worker thread to one or more of the nodes to perform data movement in response to the event.
2. The method of claim 1, wherein said worker threads are posted to one or more nodes other than said coordinating node to perform data movement tasks.
3. The method of claim 1, wherein said coordinating node is a session node.
4. The method of claim 1, further comprising providing data management access rights to the one or more nodes to which said worker threads are posted, and permitting only the one or more nodes having said data management access rights to execute said worker threads.
5. The method of claim 1, further comprising establishing a process session in said cluster and assigning a session identifier for that session.
6. The method of claim 5, further comprising providing said session identifier to said one or more nodes to which said worker threads are posted, and permitting only the one or more nodes having said session identifier to execute said worker thread.

7. The method of claim 5, wherein said DM application establishes said session and assigns said session identifier.

8. The method of claim 5, wherein a plurality of sessions are established in said cluster concurrently and each session is assigned a unique session identifier.

9. The method of claim 1, wherein said DM application utilizes one or more parallel file systems for management of data.

10. The method of claim 9, wherein each parallel file system further comprises one or more physical file systems.

11. The method of claim 10, wherein said worker threads include calls for performing at least one of punching holes in files, moving data into files and moving data out of files.

12. The method of claim 9, wherein said DM application is initiated using a data management application programming interface (DMAPI).

13. The method of claim 1, wherein said DM application is initiated using a data management application programming interface (DMAPI).

14. The method of claim 1, wherein said processing environment includes a storage area network (SAN) including said one or more data storage units.

15. The method of claim 12, wherein said processing environment includes a storage area network (SAN) including said one or more data storage units.

16. The method of claim 14, wherein said worker threads perform data movement within a hierarchical storage management (HSM) system.

17. The method of claim 1, further comprising reassigning a worker thread to another node upon failure of the node to which the worker thread is dispatched.

18. The method of claim 1, further comprising assigning another coordinating node upon failure of the coordinating node.

19. A machine readable medium having a set of instructions recorded thereon for performing a method of managing data movement, said method including:

establishing a processing environment in a cluster of nodes having common access to data residing in one or more data storage units;

initiating a data management application (DM) in said environment;

assigning a node of said cluster as a coordinating node for managing data movement;

receiving an event by the coordinating node requesting movement of data;

posting a worker thread to one or more of the nodes to perform data movement in response to the event.

20. A system for managing data movement comprising;

a computing environment having a cluster of nodes having common access to data residing in one or more data storage units;

a data management application (DM) operable to manage data movement by assigning any node in said cluster as a coordinating node to manage data movement events and dispatching worker threads to one or more nodes to perform data movement tasks in response to the data movement events.



## **IX. EVIDENCE APPENDIX**

Not applicable.

## **X. RELATED PROCEEDINGS APPENDIX**

None.